Algorithmic Confidence – A Key Criterion for XAI and FAT

Ulf Johansson
Dept. of Computer Science and Informatics
Jönköping University, Sweden,
ulf.johansson@ju.se

Cecilia Sönströd
Dept. of Information Technology
University of Borås, Sweden
cecilia.sonstrod@hb.se

Abstract

As AI is increasingly used not only for decision support, but also automated decision making, trust in the resulting decisions or recommendations becomes vital. Consequently, how to enable trust in AI solutions is today a key question addressed by researchers from many disciplines. The importance of trust in AI is also strongly manifested in the two vibrant areas Explainable AI (XAI) and Fairness, Accountability and Transparency (FAT). The purpose of this tutorial paper is two-fold: (i) to argue for that FAT and XAI should include algorithmic confidence as a requirement for accountable and explainable AI and (ii) to show that conformal prediction and Venn prediction, with their strong validity guarantees, are very good tools for building modules that measure and communicate algorithmic confidence.

1 Introduction

In recent years, there has been a huge increase, in both societal discourse and in research, regarding the impact of algorithms in general, and artificial intelligence (AI) algorithms, in particular. In machine learning (ML) research, the number of papers featuring the terms Explainable AI (XAI) and Fairness, Accountability and Transparency (FAT) has increased dramatically during the last few years. The FAT acronym is sometimes amended to include Ethics, thus becoming FATE. The issues of FAT/XAI have also attracted interest from several other academic and professional disciplines. Many publications recognize that interdisciplinary approaches are needed and combine computer science with perspectives from e.g. law [1], policy studies [2], [3] or media studies [4].

Professional associations such as the ACM [5], FAT/ML [6] and IEEE [7] have proposed guidelines and frameworks for FAT/XAI, incorporating demands to be placed on AI solutions, as well as evaluation criteria for explainability and FAT. Although not always explicitly mentioned, enabling *trust* in the systems is a strong driving force for XAI/FAT. Humans interacting with, or affected by, AI need to be able to make informed judgments about when to trust the system and when to examine a decision in detail. This is, of course, nothing new, but has been present in the AI discourse since the era of expert systems and is also prominent in recent high-impact publications within machine learning, such as the LIME framework [8].

Representing a major research initiative within FATE/XAI, the DARPA Research Programme on Explainable AI [9], launched in 2017, is aimed at developing new AI solutions that produce "more explainable models, while maintaining a high level of learning performance (e.g., prediction accuracy)". In the programme call, a three-tier framework for measuring explanation effectiveness is outlined, with the two lower levels consisting of being able to explain a model's individual decisions, and the strengths and weaknesses of the overall model, respectively. The third, and highest level of

NIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy, Montréal, Canada. This is a non-archival publication - the authors may submit revisions and extensions of this paper to other publication venues.

explainability is for the system to enable the user to identify and correct mistakes. It is noted that this goal may not be achievable with current machine learning techniques.

The FAT/ML Principles for Accountable Algorithms and a Social Impact Statement for Algorithms [6], list *responsibility*, *explainability*, *accuracy* and *auditability* as the components of an accountable algorithm. The proposal includes that a social impact statement should be developed during algorithmic design and revised when the algorithm or system is implemented, and then made public. The social impact statement includes a set of guiding questions and initial steps that organizations can take to ensure that their algorithms meet the criteria for accountability. Under accuracy, one guiding question is "How confident are the decisions output by your algorithmic system?" Proposed solutions, in the form of initial steps, include performing sensitivity analysis, developing a process to enable human correction of errors (either in data or in decisions) and determining how to communicate the uncertainty for each decision.

We agree with the above frameworks, in that demands on explainability and accountability must include some capacity for reporting uncertainty. However, to enable informed trust in the system, algorithmic ability to reason about its own competence, specifically about confidence in individual recommendations, is needed. Thus, we argue that the criteria, as formulated, would benefit from being made more explicit and precise. In particular, an essential property in a transparent, accountable and explainable algorithm is the ability to assess and clearly communicate a confidence level for each recommendation. Ideally, an algorithmic prediction or recommendation should always be accompanied by a measure of confidence that is easily understood by a human user. For this to become the norm, two things are essential:

1. existing frameworks for FAT/XAI should include algorithmic confidence as a requirement for transparent and/or explainable AI.

2. existing viable solutions to the problem of providing algorithmic confidence must be communicated to the wider research and practitioner community

To address the first issue, we propose that the questions an algorithm should be able to answer are amended to include "*How sure are you?*" Some existing machine learning techniques incorporate mechanisms that go some way towards answering this question, typically by providing some form of confidence measure, sometimes even probability estimates. Often, however, these estimates are, for different reasons, not well-calibrated, resulting in confidence measures that, in fact, become misleading. In addition, for this question to be meaningful, the confidence level needs to be provided for each individual prediction or recommendation made.

For the second issue, we argue for the fairly recently developed framework *prediction with confidence*, which equips any predictive machine learning technique with the ability to answer the "How sure are you?" question. In regression, *conformal prediction* enables the user to specify a desired significance level ϵ, and the prediction intervals produced by the algorithm are, under the standard i.i.d. assumption, mathematically guaranteed to contain the true target with the probability $1 - \epsilon$. Here, the possibility of choosing a specific significance level makes it possible to adapt the algorithm's performance to the legal, ethical or financial constraints of a particular task. Furthermore, in many situations where a decision is based on predictive modeling, using the two endpoints of the interval will produce very robust worst-case and best-case estimations.

In classification, *Venn predictors* will, also under the i.i.d. assumption, produce probabilistic predictions that are automatically perfectly calibrated, even in the small. Obviously, well-calibrated probabilistic predictors are very strong tools for automated decision making or decision support. Specifically, if combined with utilities, such predictors will produce optimal decisions, according to Bayesian decision theory.

In this tutorial paper, we will demonstrate that the prediction with confidence framework provides a very strong foundation for modules capable of answering the question "How sure are you?"

2 Prediction with confidence

In this section, we will present the two frameworks suggested for confidence prediction, i.e., *conformal prediction* for predictive regression, and *Venn predictors* for predictive classification. In the presentation, we will start with a high-level, informal, description, explaining how and why the techniques work. After that, we give more formal descriptions, and conclude with some examples.

First, it should be noted that both frameworks are used on top of arbitrary predictive models, here called *underlying models*. With this is mind, the approaches presented here are extremely general. While both conformal prediction and Venn predictors were initially suggested in a transductive setting, we in this tutorial present only the inductive versions. In the transductive setting, at least one underlying model has to be trained for every test instance. This is, of course, computationally inefficient, but it also means that there is no fixed model that can be inspected and analyzed, thus severely hampering interpretability.

2.1 Conformal prediction

Inductive conformal regression can be described and motivated in a very straightforward way. Assume that we have a trained regression model, and that we also have access to another data set (called the *calibration set*) not used for training of the model, where the true targets are known. Now, if the underlying model predicts output values for all calibration instances, we can calculate the absolute errors for these predictions, since we know the true targets. The calibration instances are then sorted, in increasing order of their absolute errors. The key idea of inductive conformal regression is that if this calibration set is i.i.d. with future test instances, we can use the absolute errors on the calibration set to provide bounds for test set errors. If we, as an example, have exactly 100 calibration instances, sorted as described above, we would expect 20% of future (test set) predictions to have an absolute error larger than the absolute error of instance number 80. Obviously, we can pick any significance level, which would correspond to just using the absolute error of another calibration instance.

Before presenting conformal regression formally, we want to highlight four things:

- All conformal regression predictions come in the form of intervals, rather than point predictions. An error in the framework is when the true target is outside the interval.
- A conformal regressor is exactly valid, i.e., the error rate, as defined above, will be exactly equal to the preset significance level, in the long run.
- Here, "in the long run" means repeating the entire procedure, not just adding more and more test instances using a fixed calibration set. All published studies, however, show empirical error rates very close to the significance levels.
- In the most basic setting, all prediction intervals are of equal width, i.e., twice the absolute error of the calibration instance corresponding to the chosen significance level. The addition of a micro-technique dubbed *normalization* (described below), however, provides two clear benefits: (i) the prediction intervals produced differ in size, such that easier instances will have smaller intervals than instances that are more difficult, thus providing additional information on a per-instance basis and (ii) the resulting prediction intervals are typically tighter on average.

All conformal predictors utilize so-called *nonconformity functions*— real-valued functions that measure the strangeness of examples $(\boldsymbol{x}, y)$. For predictive regression, the nonconformity of an example $(\boldsymbol{x}_i, y_i)$ is often defined as the absolute error

$$A\left(\boldsymbol{x}_i, y_i, h\right) = \left|y_i - h\left(\boldsymbol{x}_i\right)\right|, \tag{1}$$

where h is some predictive model providing real-valued predictions, e.g., a regression tree, or a neural network. An inductive conformal predictor (ICP) for regression [10] is constructed as follows:

1. Divide the training data Z into two disjoint subsets:
   a proper training set Z^t and a calibration set Z^c.
2. Train the underlying model h on Z^t.
3. Use the nonconformity function, e.g. Eq. 1, to measure the nonconformity of the examples in Z^c, obtaining a list of calibration scores $S = \alpha_1, ..., \alpha_q$ where $q = |Z^c|$ and S is sorted in descending order.

When a new test instance $\boldsymbol{x}_{k+1}$ arrives, a valid prediction region for that instance and a specific confidence level ϵ is constructed as follows:

1. Obtain a prediction $h(\boldsymbol{x}_{k+1})$.
2. Find the calibration score α_p where $p = \lfloor \epsilon(q+1) \rfloor$.
3. Using the (partial) inverse of the nonconformity function, obtain the largest nonconformity score that is consistent with ϵ, i.e., $A^{-1}(\alpha_p)$. This is the maximum nonconformity score for h and $\boldsymbol{x}_{k+1}$ with confidence $1-\epsilon$.

If the nonconformity function in Eq. 1 is used, the predictive step simply translates into a prediction interval for $\boldsymbol{x}_{k+1}$ being constructed as

$$\hat{Y}^{\epsilon}_{k+1} = h\left(\boldsymbol{x}_{k+1}\right) \pm \alpha_p, \tag{2}$$

since, with probability $1-\epsilon$, the underlying model h will not make an absolute prediction error greater than α_p.

Using the standard definition of a conformal regressor (Eq. 1 and 2) all prediction regions produced are of the same size, namely $2\alpha_p$. This does not reflect the fact that it is easier to confidently predict the output of some instances than for others. By using a *normalized* nonconformity function [11–13], it is possible to vary the size of the prediction regions based on the estimated difficulty of the test examples. Here, the nonconformity of an instance is defined as

$$A\left(\boldsymbol{x}_i, y_i, h\right) = \frac{\left|y_i - h\left(\boldsymbol{x}_i\right)\right|}{\sigma_i + \beta}, \tag{3}$$

where σ_i is an estimate of the difficulty of predicting the correct output of $\boldsymbol{x}_i$, and β is a sensitivity parameter. A common way of obtaining such estimates is to use an additional model g trained on the residual errors of h, and let $\sigma_i = g(\boldsymbol{x}_i)$ [11–14]. Other options include using the average error of k-nearest neighbors [14], or the standard deviation of predicted values from an ensemble [15].

The prediction intervals made using a normalized nonconformity function are then defined as

$$\hat{Y}^{\epsilon}_{k+1} = h\left(\boldsymbol{x}_{k+1}\right) \pm \alpha_p\left(\sigma_{k+1} + \beta\right) \ . \tag{4}$$

2.1.1 Mondrian conformal prediction

All guarantees for standard conformal prediction are global, i.e., they apply only on the model-level. As an example, if a regression tree is used as the underlying model, some leaves in the tree would probably have a higher error rate than ϵ, while others would have a lower error rate [16]. Using *Mondrian* conformal prediction [17], however, it becomes possible to provide localized guarantees. In practice, the feature space is divided into several disjoint subcategories $\kappa_1, ..., \kappa_m$, and then Mondrian conformal prediction will provide guarantees for each κ_j individually. In the example with conformal regression trees, the leaves of the tree provide a natural subdivision of the problem space [16]. For such a Mondrian conformal regressor, the predictions made in *each* leaf, would be correct with probability $1-\epsilon$. Or, put in another way, if the paths from the root node to the leaves are formulated as rules, all these rules will be independently valid. To construct a Mondrian conformal regression tree, the only change needed from the non-Mondrian variant is to select α_p from S_κ rather than S, where S_κ is the set of calibration examples that fall into the same leaf as the test instance $\boldsymbol{x}_{k+1}$.

It must be noted that for ICP in general, the calibration set needs to be large enough to support the chosen significance level. Since one particular calibration instance is used to determine the interval width, there must be enough calibration instances to allow a partition into an appropriate number of quantiles. As an example, $\epsilon = 0.2$ requires at least 4 instances, while $\epsilon = 0.05$ requires at least 19 instances. In general, the minimum number of required calibration instances is $\epsilon^{-1} - 1$. This becomes especially relevant for Mondrian conformal predictors, since it is necessary to retain at least $\epsilon^{-1} - 1$ calibration instances not only for the entire model, but for each separate category in order to make valid predictions. Thus, a regression tree with m leaves requires as an absolute minimum $m\epsilon^{-1} - 1$ calibration instances, which may be a substantial number for larger trees. In effect, this puts restrictions on the overall size of the underlying regression tree; the tree must be shallow enough to allow each leaf to contain an sufficient number of calibration instances.

2.1.2 Examples Conformal Prediction

Here we present some results, replicated from [18]. In that study, standard regression trees were used as underlying models, and the overall purpose was to compare different ways of producing informative and valid conformal regressors. Specifically, the goal was to conserve interpretability in conformal regression trees by requiring that all examples falling into a leaf obtains the same prediction, while still being able to use normalization.

We start by looking at some trees from the Mortgage data set. For these trees, it was ensured that all leaves included at least 100 training instances. The reason for this very high, and not optimized, value was that it will result in quite compact trees that can be easily inspected and analyzed. Starting with the standard global ICP in Fig. 1, where $\epsilon = 0.1$, we see that all intervals have the same width[1]. When applied to the test set, containing 106 instances, the error rate is 10.38% (11 test instances), i.e., very close to the significance level. Looking at the information in the curly brackets, showing the number of instances and errors, we can see that all errors are actually committed in the $[0.539, 0.839]$ leaf, where 11 of 27 instances are errors.

```
x6<10.58
|       x6<7.25
|       |       x5<4.54
|       |       |       y=[0, 0.227] {15/0}
|       |       x5>=4.54
|       |       |       y=[0.035, 0.335] {34/0}
|       x6>=7.25
|       |       x5<7.195
|       |       |       y=[0.143, 0.443] {11/0}
|       |       x5>=7.195
|       |       |       y=[0.244, 0.544] {19/0}
x6>=10.58
|       y=[0.539, 0.839] {27/11}
```

Figure 1: Standard global ICP for mortgage. $\epsilon = 0.1$

Turning to the ICP in Fig. 2 below, which was produced using the novel normalization function suggested in [18], the intervals now have different sizes. However, all 11 errors are still committed in just one leaf, i.e., as expected, the normalized approach lead to a more efficient, but still valid, conformal regressor.

```
x6<10.58
|       x6<7.25
|       |       x5<4.54
|       |       |       y=[0, 0.216] {15/0}
|       |       x5>=4.54
|       |       |       y=[0.050, 0.320] {34/0}
|       x6>=7.25
|       |       x5<7.195
|       |       |       y=[0.157, 0.429] {11/0}
|       |       x5>=7.195
|       |       |       y=[0.257, 0.531] {19/0}
x6>=10.58
|       y=[0.539, 0.839] {27/11}
```

Figure 2: Normalized local ICP for mortgage. $\epsilon = 0.1$

For a Mondrian ICP, the error rate is bounded for each leaf, leading to tighter intervals for easier parts of the feature space and wider intervals for harder parts. Fig. 3 shows the Mondrian model for $\epsilon = 0.1$. Here, we see that the interval that contained all the errors for the other two approaches is now much larger. This Mondrian model commits altogether 8 errors, but no more than 3 errors in any specific leaf, i.e., with Mondrian ICP, the interval sizes are adjusted so that each leaf is independently valid.

[1] In this and following examples, the outputs are restricted to the interval $[0.0 - 1.0]$, so one of the intervals is actually smaller than the others.

```
x6<10.58
|       x6<7.25
|       |       x5<4.54
|       |       |       y=[0.011, 0.144] {15/1}
|       |       x5>=4.54
|       |       |       y=[0.144, 0.226] {34/3}
|       x6>=7.25
|       |       x5<7.195
|       |       |       y=[0.231, 0.355] {11/1}
|       |       x5>=7.195
|       |       |       y=[0.332, 0.456] {19/1}
x6>=10.58
|       y=[0.468, 0.910] {27/2}
```

Figure 3: Mondrian ICP for mortgage. $\epsilon = 0.1$

Table 1 below shows aggregated results, over 21 data sets, from [18]. For more detailed results, i.e., on data set level, and for a thorough description of the setups evaluated, see the original paper. The three setups tabulated here are a standard ICP (ICP), an ICP using a novel normalization function, based on the variance of the true targets in each leaf (ICP-N), and a Mondrian approach (Mon), where each leaf is a separate Mondrian category. The results show that all setups are empirically valid, with error rates exceptionally close to the significance levels. In addition, it is obvious that the normalized approach produced tighter intervals on average. Using a Mondrian approach, finally, will result in larger intervals, but of course, in those models, validity is guaranteed for every single leaf.

Table 1: Validity and Efficiency

	$\epsilon = 0.01$			$\epsilon = 0.05$			$\epsilon = 0.1$			$\epsilon = 0.2$		
	ICP	ICP-N	Mon	ICP	ICP-N	Mon	ICP	ICP-N	Mon	ICP	ICP-N	Mon
Error rate	.010	.010	.010	.050	.050	.051	.100	.100	.103	.198	.198	.200
Interval size	.518	.501	.643	.357	.337	.373	.285	.270	.292	.210	.204	.221

2.1.3 A note on conformal classification

Conformal prediction is also defined for classification; in fact, most published papers on conformal prediction investigate classification, using different underlying models, e.g., ANNs [19], kNNs [20], SVMs [21, 22], decision trees [23], random forests [21, 24] and evolutionary algorithms [25, 26].

In conformal classification [17], each prediction consists of a subset of the labels, and the probability of excluding the true class label is bounded by the predefined significance level.

We must however, be very careful when interpreting conformal classifiers. We know that we will make exactly ϵ errors in the long run, and that an error is when the correct label is not in the predicted label set. With this in mind, the guarantee really only applies *a priori*, i.e., once we have seen a specific prediction, we cannot say that the probability for that prediction to be wrong is ϵ. As an example, consider a two-class problem. Here, a number of instances are likely to obtain prediction sets containing both classes, meaning that these instances cannot be erroneous. Thus, all errors must be made on the remaining singleton predictions. So, once we have observed a singleton prediction, the probability for that being incorrect is most likely much higher than ϵ. We argue that this makes conformal classifiers hard to use in practice. With this in mind, we instead recommend Venn predictors for classification with confidence.

2.2 Venn prediction

Venn predictors [27], are multi-probabilistic predictors with proven validity properties. The general impossibility result regarding validity for probabilistic prediction is circumvented in two ways: (i) multiple probabilities for each label are output, with one of them being the valid one and (ii) the statistical tests for validity are restricted to calibration. In practice, the probabilities should be matched by observed frequencies. As an example, if we make a number of predictions with the probability estimate 0.9, we expect these predictions to be correct in about 90% of the cases.

Similarly to conformal regression, Venn predictors also use underlying models and calibration sets with known labels. The key idea of Venn prediction is to somehow use the underlying model to divide all calibration examples into a number of *categories*. When performing predictions, the underlying model is used, in exactly the same way as for the calibration instances, to determine the category for each test instance. Then, the estimated class probabilities for a test instance is calculated from the relative frequencies of the labels for the calibration instances belonging to that category. To obtain validity, this calculation must include the test instance to be predicted. However, since the true label is, per definition, not known for the test instance, all possible labels must be considered, which results in a set of C label probability distributions, where C is the number of possible labels. We now describe inductive Venn prediction:

Assume we have a training set of the form $\{z_1, \ldots, z_l\}$ where each instance $z_i = (x_i, y_i)$ consists of two parts; an *object* x_i and a *label* y_i. Divide this training set into the proper training set $\{z_1, \ldots, z_q\}$ and the calibration set $\{z_{q+1}, \ldots, z_l\}$. When presented with a new test object x_{l+1}, Venn prediction estimates the probability that $y_{l+1} = Y_j$, for each Y_j in the set of possible labels $Y_j \in \{Y_1, \ldots, Y_c\}$. Here, the calibration examples are divided into a number of *categories* and then the relative frequency of label $Y_j \in \{Y_1, \ldots, Y_c\}$ in each category is used to estimate label probabilities for test instances falling into that category. The categories are defined using a *Venn taxonomy* where different taxonomies lead to different Venn predictors. Normally, the taxonomy is based on the underlying model, i.e., for each calibration and test object x_i, the output of the underlying model is somehow used to assign (x_i, y_i) into one of the categories. The most basic Venn taxonomy, used in for instance [28], simply puts all examples predicted with the same label into the same category.

The result of Venn prediction of a test instance x_{l+1}, is a set of label probability distributions. Instead of dealing directly with these distributions, an often employed compact representation is to use the lower $L(Y_j)$ and upper $U(Y_j)$ probability estimates for each label Y_j. Let k be the category assigned to the test object x_{l+1} by the Venn taxonomy, and Z_k be the set of calibration instances belonging to category k. Then the lower and upper probability estimates are defined by:

$$L(Y_j) = \frac{|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}|}{|Z_k| + 1} \tag{5}$$

and:

$$U(Y_j) = \frac{|\{(x_m, y_m) \in Z_k \mid y_m = Y_j\}| + 1}{|Z_k| + 1} \tag{6}$$

In order to make a prediction $\hat{y}_{l+1}$ for x_{l+1} using the lower and upper probability estimates, the following procedure is employed:

$$\hat{y}_{l+1} = \max_{Y_j \in \{Y_1, \ldots, Y_c\}} L(Y_j) \tag{7}$$

The output of a Venn predictor is the above prediction $\hat{y}_{l+1}$ together with the probability interval:

$$[L(\hat{y}_{l+1}), U(\hat{y}_{l+1})] \tag{8}$$

It is proven that the multiprobability predictions produced by Venn predictors are automatically valid, regardless of the taxonomy used [29]. Still, the taxonomy is important, since it will affect both the accuracy of the Venn predictor and the size of the prediction interval. Obviously, smaller probability intervals are more informative, and the probability estimates should preferably be as close to one or zero as possible. In addition, it must be noted that the more categories that are used in the taxonomy, the more specific the predictions will be. As an example, if we consider a two-class problem, the basic taxonomy that puts all the examples predicted with the same label into the same category will have exactly two categories, so the Venn predictor will for every test instance output one of only two possible prediction intervals. However, if we have too many categories, the calibration will depend on just a few instances, resulting in larger intervals.

2.2.1 Venn-Abers predictors

As described above, the key challenge of Venn predictors is to identify the most suitable taxonomy to use. Venn-Abers predictors [30] are Venn predictors applicable to two-class problems, where the taxonomy is automatically optimized using isotonic regression. Thus, the Venn-Abers predictor inherits the validity guarantee of Venn predictors, while providing specific predictions.

Since Venn-Abers predictors are restricted to two-class problems, they can regard the underlying models as *scoring classifiers*, i.e., when an underlying model makes a prediction for a test object, the output is a *prediction score* $s(x)$, where a higher value indicates a larger belief in that the test instance has the label 1. For scoring classifiers applied to two-class problem (using labels 0 and 1) the actual prediction is normally obtained by comparing the score to a fixed threshold c, and predicting the label of x to be 1 if $s(x) > c$, and 0 otherwise. An alternative to using a fixed threshold c is, however, to first somehow calibrate an increasing function g using a number of predictions scores, with known true targets. After such a calibration, $g(s(x))$ should be interpreted as the probability that the label for x is 1.

Venn-Abers predictors use isotonic regression for the calibration. A multi-probabilistic prediction from a Venn-Abers predictor is, in the inductive setting, produced as follows; let s_0 be the scoring function for $\{z_{q+1}, \dots, z_l, (x_{l+1}, 0)\}$, s_1 be the scoring function for $\{z_{q+1}, \dots, z_l, (x_{l+1}, 1)\}$, g_0 be the isotonic calibrator for

$$\{(s_0(x_{q+1}), y_{q+1}), \dots, (s_0(x_l), y_l), (s_0(x_{l+1}), 0)\} \tag{9}$$

and g_1 be the isotonic calibrator for

$$\{(s_1(x_{q+1}), y_{q+1}), \dots, (s_1(x_l), y_l), (s_1(x_{l+1}), 1)\} \tag{10}$$

Then the probability interval for $y_{l+1} = 1$ is

$$[g_0(s_0(x_{l+1})), g_1(s_1(x_{l+1}))] \tag{11}$$

2.2.2 Examples Venn Prediction

Fig. 4 below shows a probability estimation tree induced on the Wisconsin breast cancer data set, and then calibrated using Venn-Abers. Again, the settings were chosen to produce extremely small trees. Here, we see that the probability intervals differ substantially in size. The two most common leaves have very tight probability intervals, and appear to be well-calibrated from the observed error rates, even for this single test fold consisting of 47 test instances. The other two leaves have much larger probability intervals, showing that the model is less confident, simply because these leaves have much fewer observations.

```
x2<3.5
|       x6<4.5
|       |   y= Class 0 [0.9833, 1.0000] {22/0}
|       x6>=4.5
|       |   x3<1.5
|       |   |   y= Class 0 [0.5000, 0.6000] {1/0}
|       |   x3>=1.5
|       |   |   y= Class 1 [0.8000, 0.9091] {2/1}
x2>=3.5
|       y= Class 1 [0.8864, 0.9091] {22/2}
```

Figure 4: Venn-Abers PET, WBC dataset

Table 2 below shows aggregated results, over 25 data sets, from a work-in-progress study. Here, Venn-Abers predictors were compared to using a Laplace correction, as well as the two standard techniques Platt scaling [31] and isotonic regression [32] for calibrating probability estimation trees. Interestingly enough, the Venn-Abers obtained significantly lower log losses and Brier losses, compared to all other methods.

Table 2: Log loss and Brier loss

	Log Loss				Brier Loss			
	LaP	Platt	Iso	VAP	LaP	Platt	Iso	VAP
Mean	.795	.707	∞	.681	.175	.160	.157	.155
Mean rank	2.80	2.12	4.00	1.08	3.52	3.04	2.28	1.16

References

[1] T. Zarsky, “The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making,” *Science, Technology, & Human Values*, vol. 41, no. 1, pp. 118–132, 2016.

[2] J. Zerilli, A. Knott, J. Maclaurin, and C. Gavaghan, “Transparency in algorithmic and human decision-making: Is there a double standard?” *Philosophy & Technology*, Sep 2018. [Online]. Available: https://doi.org/10.1007/s13347-018-0330-6

[3] J. Danaher, M. J. Hogan, C. Noone, R. Kennedy, A. Behan, A. D. Paor, H. Felzmann, M. Haklay, S.-M. Khoo, J. Morison, M. H. Murphy, N. O'Brolchain, B. Schafer, and K. Shankar, “Algorithmic governance: Developing a research agenda through the power of collective intelligence,” *Big Data & Society*, vol. 4, no. 2, p. 2053951717726554, 2017.

[4] M. Ananny and K. Crawford, “Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability,” *New Media & Society*, vol. 20, no. 3, pp. 973–989, 2018.

[5] “New statement on algorithmic transparency and accountability by ACM U.S. Public Policy Council,” Association for Computing Machinery, 2017. [Online]. Available: https://techpolicy.acm.org/?p=6156

[6] N. Diakopoulos, S. Friedler, M. Arenas, S. Barocas, M. Hay, B. Howe, H. V. Jagadish, K. Unsworth, A. Sahuguet, S. Venkatasubramanian, C. Wilson, and B. Z. C. Yu, *Principles for Accountable Algorithms and a Social Impact Statement for Algorithms*, FAT/ML, 2017. [Online]. Available: http://www.fatml.org/resources/principles-for-accountable-algorithms

[7] “The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems,” IEEE, 2017. [Online]. Available: http://standards.ieee.org/develop/indconn/ec/autonomous_systems.html

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778

[9] “Explainable Artificial Intelligence (XAI),” Defense Advanced Research Projects Agency. Broad Agency Announcement, DARPA-BAA-16-53, 2016. [Online]. Available: https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf

[10] H. Papadopoulos, “Inductive conformal prediction: Theory and application to neural networks,” *Tools in Artificial Intelligence*, vol. 18, pp. 315–330, 2008.

[11] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, “Inductive confidence machines for regression,” in *Machine Learning: ECML 2002*. Springer, 2002, pp. 345–356.

[12] H. Papadopoulos and H. Haralambous, “Neural networks regression inductive conformal predictor and its application to total electron content prediction,” in *Artificial Neural Networks – ICANN 2010*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6352, pp. 32–41.

[13] ——, “Reliable prediction intervals with regression neural networks,” *Neural Networks*, vol. 24, no. 8, pp. 842–851, 2011.

[14] U. Johansson, H. Boström, T. Löfström, and H. Linusson, “Regression conformal prediction with random forests,” *Machine Learning*, vol. 97, no. 1-2, pp. 155–176, 2014.

[15] H. Boström, H. Linusson, T. Löfström, and U. Johansson, “Accelerating difficulty estimation for conformal regression forests,” *Annals of Mathematics and Artificial Intelligence*, vol. 81, no. 1-2, pp. 125–144, 2017.

[16] U. Johansson, C. Sönströd, H. Boström, and H. Linusson, "Regression trees for streaming data with local performance guarantees," in *IEEE International Conference on Big Data*. IEEE, In press, 2014.

[17] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005.

[18] U. Johansson, H. Linusson, T. Löfström, and H. Boström, "Interpretable regression trees using conformal prediction," *Expert Syst. Appl.*, vol. 97, pp. 394–404, 2018.

[19] H. Papadopoulos, "Inductive conformal prediction: Theory and application to neural networks," *Tools in artificial intelligence*, vol. 18, no. 315-330, p. 2, 2008.

[20] K. Nguyen and Z. Luo, "Conformal prediction for indoor localisation with fingerprinting method," *Artificial Intelligence Applications and Innovations*, pp. 214–223, 2012.

[21] D. Devetyarov and I. Nouretdinov, "Prediction with confidence based on a random forest classifier," *Artificial Intelligence Applications and Innovations*, pp. 37–44, 2010.

[22] L. Makili, J. Vega, S. Dormido-Canto, I. Pastor, and A. Murari, "Computationally efficient svm multi-class image recognition with confidence measures," *Fusion Engineering and Design*, vol. 86, no. 6, pp. 1213–1216, 2011.

[23] U. Johansson, H. Boström, and T. Löfström, "Conformal prediction using decision trees," in *International Conference Data Mining (ICDM)*. IEEE, 2013.

[24] S. Bhattacharyya, "Confidence in predictions from random tree ensembles," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 71–80.

[25] U. Johansson, R. König, T. Löfström, and H. Boström, "Evolved decision trees as conformal predictors," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 1794–1801.

[26] A. Lambrou, H. Papadopoulos, and A. Gammerman, "Reliable confidence measures for medical diagnosis with evolutionary algorithms," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 1, pp. 93–99, 2011.

[27] V. Vovk, G. Shafer, and I. Nouretdinov, "Self-calibrating probability forecasting," in *Advances in Neural Information Processing Systems*, 2004, pp. 1133–1140.

[28] U. Johansson, T. Löfström, H. Sundell, H. Linusson, A. Gidenstam, and H. Boström, "Venn predictors for well-calibrated probability estimation trees," in *COPA*. PMLR, 2018, pp. 1–12.

[29] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer-Verlag New York, Inc., 2005.

[30] V. Vovk and I. Petej, "Venn-abers predictors," *arXiv preprint arXiv:1211.0025*, 2012.

[31] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[32] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proc. 18th International Conference on Machine Learning*, 2001, pp. 609–616.